

FACULDADE FACIMP WYDEN

SISTEMAS OPERACIONAIS

Brenno Mendes Sousa – 202303654995

Kaio Mateus Silva Sousa – 202302364462

José Guilherme da Silva Pinho – 202302698735

ATIVIDADE

Objetivo: Compreender o funcionamento do BCP e do Scheduler em um sistema operacional, representando graficamente o ciclo de vida de um processo e a interação entre o BCP e o Scheduler.

1. Pesquisa e Resumo (BCP)

1.1 O que é o BCP (Bloco de Controle de Processos)?

É uma estrutura de dados fundamental em sistemas operacionais que contém a informação a cerca de um processo.

1.2 Quais informações o BCP armazena?

Ele armazena informações essenciais para o gerenciamento do processo pelo sistema operacional. Algumas informações típicas armazenadas são:

- Identificação do Processo (PID – Process ID);
- Estado do Processo (executando, pronto, bloqueado);
- Contador de Programa;
- Registros da CPU;
- Informações de Gerenciamento de Memória;
- Informações de Contabilidade;
- Informações de Gerenciamento de Arquivos;
- Informações de Gerenciamento de Recursos e Estado de E/S;
- Ponteiros para outros Processos.

1.3 Qual é o papel do BCP no gerenciamento de processos?

O BCP é a principal estrutura usada pelo sistema operacional para manter o controle sobre os processos e garantir que eles sejam executados de maneira eficiente e ordenada. Os principais papéis do BCP no gerenciamento de processos:

- **Armazenamento de Estado do Processo:** O BCP armazena o estado atual do processo (como executando, pronto, bloqueado), permitindo que o sistema operacional saiba o que fazer com o processo em cada momento. Esse armazenamento é essencial para o escalonamento e para a troca de contexto entre processos.
- **Facilitação da Troca de Contexto (Context Switch):** A troca de contexto ocorre quando o sistema operacional interrompe um processo em execução e começa a executar outro. Durante essa troca, o estado do processo interrompido é salvo no seu BCP, incluindo o contador de programa (endereço da próxima instrução a ser executada) e os registros da CPU. Quando o processo é retomado, o sistema operacional restaura seu estado a partir do BCP, permitindo que ele continue de onde parou.
- **Gerenciamento de Recursos:** O BCP mantém informações sobre os recursos que um processo está utilizando ou aguardando, como arquivos abertos, dispositivos de entrada/saída, e áreas de memória. Isso ajuda o sistema operacional a alocar e liberar recursos de maneira eficiente e a evitar conflitos entre processos que possam estar tentando acessar os mesmos recursos simultaneamente.
- **Informações de Controle e Contabilidade:** Ele armazena informações de contabilidade, como o tempo de CPU utilizado por um processo, prioridade, e outros dados que são usados para tomada de decisão no escalonamento de processos e para garantir um uso justo e eficiente dos recursos do sistema.
- **Suporte ao Escalonamento de Processos:** Com base nas informações do BCP, o escalonador do sistema operacional decide qual processo será executado em seguida. O escalonador pode usar critérios como prioridade, tempo de espera, e outros fatores armazenados no BCP para tomar decisões de escalonamento.

- **Gerenciamento de Memória:** O BCP também contém informações relacionadas à memória que o processo está utilizando, como ponteiros para tabelas de páginas ou segmentos de memória. Essas informações são necessárias para o sistema operacional gerenciar a memória virtual e garantir que cada processo tenha acesso apenas às suas próprias áreas de memória.
- **Comunicação entre Processos:** Em alguns sistemas operacionais, o BCP pode conter informações usadas para a comunicação entre processos (IPC), como semáforos, filas de mensagens ou outros mecanismos de sincronização e comunicação.

2. Pesquisa e Resumo (Scheduler)

2.1 Os tipos de algoritmos de escalonamento (ex. FIFO, Round Robin, Prioridade)

Os algoritmos de escalonamento (ou *schedulers*) são responsáveis por decidir a ordem em que os processos serão executados pela CPU em sistemas operacionais multitarefa. Cada algoritmo possui suas características e é escolhido com base nos requisitos de desempenho, eficiência e objetivos específicos do sistema. Alguns tipos de escalonamento:

- **FIFO (First-In, First-Out):** O algoritmo FIFO é o mais simples dos algoritmos de escalonamento. Os processos são atendidos na ordem em que chegam na fila de prontos, ou seja, o primeiro processo a entrar é o primeiro a ser executado até a sua conclusão.
 - Vantagens: Fácil de implementar e justo em termos de ordem de chegada.
 - Desvantagens: Pode causar o problema de "convoy effect", onde processos curtos ficam presos atrás de processos longos, aumentando o tempo de espera médio, e, não é preemptivo, o que significa que um processo em execução não é interrompido até que ele termine.
- **Round Robin (RR):** O algoritmo Round Robin é um dos mais utilizados em sistemas de tempo compartilhado. Cada

processo recebe uma pequena quantidade de tempo da CPU (chamado de *quantum*) e é colocado no final da fila se não terminar durante seu quantum. Se um processo é interrompido (preempção) após o término do quantum, ele é colocado no final da fila de prontos.

- Vantagens: Justo, pois todos os processos recebem uma quantidade de tempo igual a CPU e boa resposta para sistemas interativos, já que processos são frequentemente alternados.
- Desvantagens: O desempenho pode ser afetado se o quantum for muito pequeno ou muito grande.

- **Escalonamento por Prioridade:** Cada processo é atribuído uma prioridade e o escalonador seleciona o processo com a prioridade mais alta para ser executado. Este tipo de escalonamento pode ser preemptivo ou não preemptivo. Em sistemas preemptivos, se um novo processo com prioridade mais alta chega, ele pode interromper o processo em execução.

- Vantagens: Flexível, permitindo que os processos mais importantes sejam atendidos primeiro.
- Desvantagens: Necessita de um mecanismo para gerenciar e ajustar prioridades para evitar a inanção.

- **Escalonamento de Curto Prazo com feedback:** Este algoritmo usa múltiplas filas com diferentes prioridades. Os processos podem ser movidos entre filas com base em seu comportamento e tempo de execução. Um processo que usa muita CPU pode ser movido para uma fila de menor prioridade, enquanto processos que esperam por I/O podem ser promovidos a filas de maior prioridade.

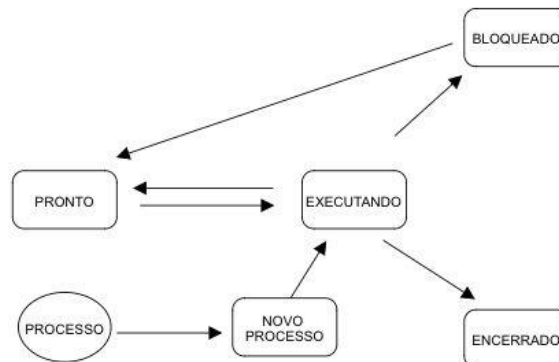
- Vantagens: Muito flexível e pode ser ajustado para diferentes tipos de cargas de trabalho e melhor uso do tempo da CPU, ajustando dinamicamente as prioridades dos processos.
- Desvantagens: Complexo de implementar e ajustar corretamente e pode ser difícil prever o comportamento do sistema devido à sua natureza dinâmica.

2.2 Como Scheduler decide qual processo deve ser executado a seguir.

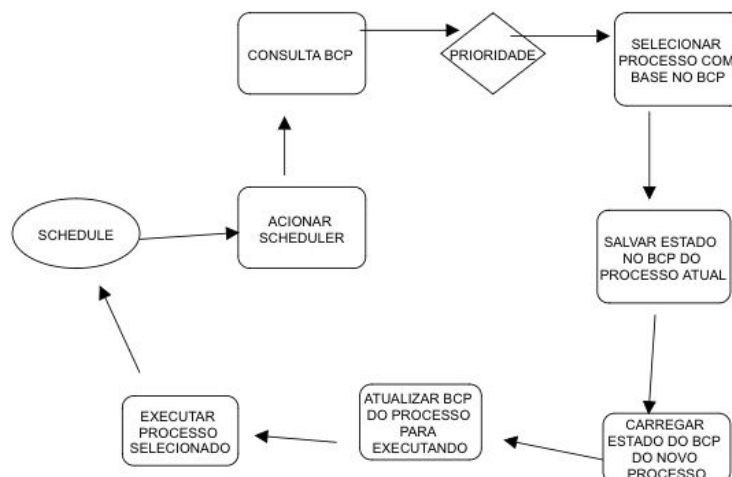
O Scheduler decide qual processo deve ser executado a seguir com base em critérios específicos, que variam conforme o algoritmo de escalonamento em uso. Por exemplo: FIFO (First-In, First-Out) o escalonador escolhe o processo que chegou primeiro na fila de prontos. Os processos são atendidos na ordem de chegada, sem preempção. O próximo processo a ser executado é simplesmente o primeiro na fila.

3. Fluxogramas

3.1 Processos e suas etapas (“Pronto”, “Executando” e “Bloqueado”)



3.2 Interação entre BCP e Scheduler.



4. Conclusão

Ao discutir os desafios e a importância de um gerenciamento eficiente de processos, é essencial considerar tanto o papel do BCP (Bloco de Controle de Processos) quanto o do *scheduler* (escalonador). Além disso, entender como diferentes algoritmos de escalonamento podem impactar o desempenho do sistema operacional é fundamental para analisar como o sistema gerencia múltiplos processos de maneira eficaz.